VŠB-Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra 460 - Katedra informatiky

Analyzátor sběrnice 1-wire

1-Wire Bus Analyzer

Akademický rok 2011/2012 **Lukáš Zajac**

Zadání

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Lukáš Zajac**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Analyzátor sběrnice 1-wire
1-Wire Bus Analyzer

Zásady pro vypracování:

Cílem práce je návrh HW modulu pro převod dat sběrnice 1-wire do PC a napsání softwaru pro PC, který bude získaná data zobrazovat, analyzovat a interpretovat.

1. Navrhněte rozhraní sběrnice 1-wire a USB.
2. Napište program pro zachytávání dat pomocí tohoto přípravku.
3. Program musí identifikovat reset, odpověď a data.
4. Pomocí textové databáze musí umožnit identifikovat různé prvky sběrnice včetně registrů.
5. Zdrojové kódy jsou součástí práce.

Seznam doporučené odborné literatury:

T.V. Sickle, Programming microcontrollers in C, Elsevier 2003, ISBN 1-878707-98-1
A. F. Vandome, F. P. Miller, J. McBrewster, 1-Wire, VDM Publishing House, ISBN 9786130743536
Český popis 1-wire sběrnice, http://www.hw.cz/rozhrani/art1215-sbernice-1-wire.html

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Michal Jahelka, Ph.D.**

Datum zadání: 18.11.2011
Datum odevzdání: 04.05.2012

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 4.5.2012

Lukáš Zajac

.....

podpis studenta

Poděkování

Tímto děkuji svému vedoucímu bakalářské práce, Ing. Michalu Jahelkovi, Ph.D., za rady, připomínky a náměty. Rovněž mu děkuji za konzultace mimo období výuky.

Abstrakt a klíčová slova

Abstrakt:

Cílem práce je vytvořit analyzátor sběrnice 1-Wire, navržené firmou Dallas Semiconductor. Výsledný produkt komunikuje s počítačem skrze rozhraní USB. Práce obsahuje samotný návrh hardwarového modulu, program pro použitý mikropočítač a obslužnou aplikaci bežící v počítači. Uživatel pomocí počítačové aplikace ovládá hardwarový modul, který umožňuje odposlouchávat nebo komunikovat s 1-Wire sběrnicí. Aplikace v počítači data vyhodnocuje, dekóduje a zobrazuje v uživatelsky přijatelné podobě. Hardwarový modul se zakládá na mikropočítači firmy Microchip a převodníku z RS232 do USB FT232RL. Firmware mikropočítače je vytvořen za použití vývojových nástrojů společnosti Microchip a obslužná aplikace využitím platformy Java.

Klíčová slova:

1-Wire, iButton, DS18B20, Dallas Semiconductor, JAVA, Microchip, FTDI, FT232RL, PIC18F15K50

Abstract:

The aim is to create analyzer 1-Wire bus, designed by Dallas Semiconductor. The resulting product communicates with the computer through the USB interface. The work itself contains a hardware module design, program used for microcomputer and service application running on your computer. The user using the computer application to controls the a hardware module, that allows you to monitor or communicate with a 1-Wire bus. Application on the computer evaluates the data, decodes and displays in a user-acceptable form. The hardware module is based on Microchip's microcomputer and converter from RS232 to USB FT232RL. Firmware of microcomputer is designed using Microchip's development tools and service application using Java platform.

Key words:

1-Wire, iButton, DS18B20, Dallas Semiconductor, JAVA, Microchip, FTDI, FT232RL, PIC18F15K50

Seznam zkratek a symbolů

CRC	Cyclic redundancy check (Cyklický redundantní součet), hašovací funkce, používaná k detekci chyb během přenosu či ukládání dat
DIP	dual in-line package, druh pouzdra integrovaného obvodu
EEPROM	Electrically Erasable Programmable Read-Only Memory, paměť jejíž obsah se dá mazat elektricky
EPROM	Erasable Programmable Read-Only Memory, paměť jejíž obsah se dá mazat UV zářením
EUSART	Enhanced Universal Synchronous Asynchronous Receiver Transmitter, sériová vstupně/výstupní periferie
I2C	Inter-Integrated Circuit, multi-masterová počítačová sériová sběrnice vyvinutá firmou Philips
ICSP	In Circuit Serial Programming (seriové programování v obvodu), metoda přímého programování mikroprocesorů
IDE	Integrated Development Environment (vývojové prostředí), software k usnadnění práce programátorů
JDK	Java Development Kit (Java vývojová sada), soubor základních nástrojů pro vývoj aplikací pro platformu Java
JRE	Java Runtime Environment (běhové prostředí Java), nutné pro spouštění aplikací v jazyce Java a i pro běh vývojových nástrojů
LED	Light-Emitting Diode (dioda emitující světlo), polovodičová součástka vyzařující světlo
PC	Personal Computer (osobní počítač)
PWM	Pulse Width Modulation (Pulsně šířková modulace), přenos analogového signálu pomocí dvouhodnotového signálu
QFN	quad-flat no-leads, druh pouzdra integrovaného obvodu
RISC	Reduced Instruction Set Computer (mikroporcesor s redukovanou instrukční sadou), architektura mikroprocesorů
ROM	Read-only Memory, pamět pouze ke čtení
RS-232	Standard sériové linky z roku 1969, komunikační rozhraní osobních počítačů a další elektroniky
SMD	surface mount device, součástka pro povrchovou montáž
SPI	Serial Peripheral Interface (sériové periferní rozhraní), komunikace mezi řídícími mikroprocesory a ostatními integrovanými obvody (EEPROM, A/D převodníky, displeje…)
USB	Universal Serial Bus (univerzální sériová sběrnice), je sériová sběrnice určená primárně pro připojeni periferií k PC
XML	Extensible Markup Language (rozšiřitelný značkovací jazyk), obecný značkovací jazyk pro výměnu dat mezi aplikacemi
XOR	exclusive or, exkluzivní logický součet

Obsah

Obsah

1 Úvod

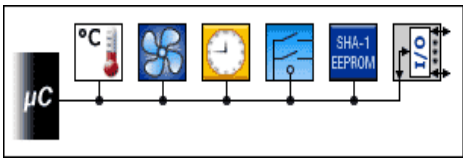
Tato práce si klade za cíl vytvořit hardwarový modul a jeho obslužnou aplikaci pro PC, která umožňuje komunikaci mezi PC a 1-wire zařízeními a odposlech komunikace na 1-wire sběrnici. Zařízení komunikuje s PC pomocí rozhraní USB. Návrh hardwarového modulu je proveden v programu CadSoft Eagle a při jeho konstrukci je využito převážně SMD součástek. Základním prvkem celého hardwarového modulu je mikropočítač PIC18F14K50 a převodník RS232 na USB FT232RL. Firmware mikropočítače je vytvořen za použití vývojových nástrojů firmy Microchip. Poslední a nejpodstatnější část je aplikace běžící v PC, která komunikuje s hardwarovým modulem a má za úkol vyhodnocovat a zpracovávat data zaslaná modulem. Aplikace je naprogramována v jazyce Java za využití volně dostupného vývojového prostředí NetBeans IDE.

2 Sběrnice 1-Wire

Sběrnice 1-wire byla vytvořena firmou Dallas Semiconductors. Tato sběrnice umožňuje připojit k řídícímu zařízení (master) pomocí pouhých dvou vodičů (datový vodič, zem) jedno, či až několik podřízených zařízení (slave). Tato technologie se využívá pro komunikaci s malými zařízeními jako jsou čidla teploty, paměti (EEPROM, EPROM,...), hodiny, napěťové a proudové senzory, hlídače baterií, iButtony a jiná další.

2.1 Obecně o sběrnici a zařízeních

Jak již vyplývá z názvu 1-wire (v překladu 1-vodič), tak se pro přenos dat používá pouze jeden vodič a společná zem. Jelikož zařízení nemají připojeno napájení (neplatí pro všechna), je v nich umístěn kondenzátor s kapacitou 800pF-2000pF, který se stará o napájení zařízení v době, kdy se datový vodič používá pro přenos dat.



Obrázek 1: 1-wire architektura

1-wire zařízení může být jedna z mnoha součástek umístěných na tištěném spoji, ale taktéž se vyskytuje i samostatně, jako například teplotní záznamník, nebo je přímo připojen k monitorovanému zařízení.
Senzorový systém je tvořen propojením několika 1-wire zařízení, s nichž každé musí znát potřebnou logiku komunikace po sběrnici. Tento systém může být připojen k počítači za použití převodníku do USB, RS232 nebo paralelního portu, nebo je propojen s nějakým mikrokontrolérem (Microchip PIC, Atmel....).

Obrázek 2: iButton

Speciálním skupinou zařízení využívající 1-wire sběrnici pro komunikaci jsou iButtony. Jsou to zařízení umístěné v malém ocelovém „knoflíku“, připomínající baterii do hodinek, který zajišťuje jejich odolnost vůči vnějším vlivům. Tyto zařízení se nejčastěji využívají jako autorizační čipy (elektronické zámky, parkovací hodiny, softwarová autorizace), ale existuje i záznamník teplot a mnoho dalších variant.

2.2 Sběrnicový systém 1-wire:

1-wire sběrnice má vždy jedno master zařízení a jeden, nebo několik slave zařízení. Například DS18B20 je vždy slave zařízení. Pokud je na sběrnici pouze jedno slave zařízení, jedná se o takzvaný „single-drop“ , pokud více slave zařízení, tak „multi-drop“.

2.3 Hardwarová konfigurace:

Sběrnice 1-wire má podle definice jeden datový vodič. Ať master, nebo slave zařízení, komunikuje s datovou linkou pomocí portu s otevřeným kolektorem, nebo 3-stavovým portem. Toto umožňuje zařízení odpojit se od sběrnice a umožnit vysílaní dalším zařízením.

Ke sběrnici 1-wire musí být připojen externí pullup rezistor s hodnotou okolo 5kΩ. Sběrnice je ve stavu nečinnosti v logické jedničce. Pokud musí být transakce z jakéhokoliv důvodu přerušena, musí být sběrnice ponechána v logické jedničce, aby mohla následně transakce plynule pokračovat. Pokud je sběrnice připojena k zemi déle než 480us, dojde k resetu všech připojených zařízení.

2.4 Transakční sekvence:

Každá transakce probíhá přesně v těchto krocích:

1. Inicializace
2. ROM příkaz
3. Funkční příkaz

Zařízení nebude správně komunikovat, pokud nějaký z kroků vynecháme, nebo provedeme v rozdílném pořadí. Vyjímku tvoří pouze dva ROM příkazy a to Search ROM a Alarm Search, po každém z těchto dvou příkazů se master musí vrátit ke kroku jedna.

2.4.1 Inicializace:

Všechny transakce na sběrnici začínají inicializační sekvencí. Inicializační sekvence se skládá z resetovacího pulzu vyvolaného masterem následovaného prezenčním pulzem vyslaným slavem/vy.

2.4.2 ROM příkazy:

Po přijetí prezenčního pulzu může master začít používat ROM příkazy. Tyto příkazy využívají unikátní 64bitový ROM kód z každého zařízení přítomného na sběrnici a umožňují specifikaci konkrétního zařízení, pokud jich je na sběrnici umístěno více. Pomocí těchto příkazů může master zjistit počet zařízení na sběrnici a jejich typ, nebo zda nějaké zařízení vyvolalo alarm. Master musí vždy použít vhodný ROM příkaz, před použitím funkčních příkazů.

2.4.3 Funkční příkazy

Po zaslání ROM příkazu a výběru cílového slave zařízení na řadu přichází použití funkčních příkazů. Tyto příkazy umožňují masterovi zapisovat a číst zápisníkovou paměť, zahájit teplotní měření, zjistit typ napájení a další podle možností připojeného zařízení.

2.5 Komunikační protokol:

Komunikace probíhá sériově, čili se data posílají postupně bit po bitu. Komunikační protokol je založen na několika typech signálů. Reset, prezenční pulz, read time sloty a write time sloty. Každý z těchto signálů má pevně daný průběh, který je nutno dodržet.

Protokol nabízí dvě komunikační rychlosti:

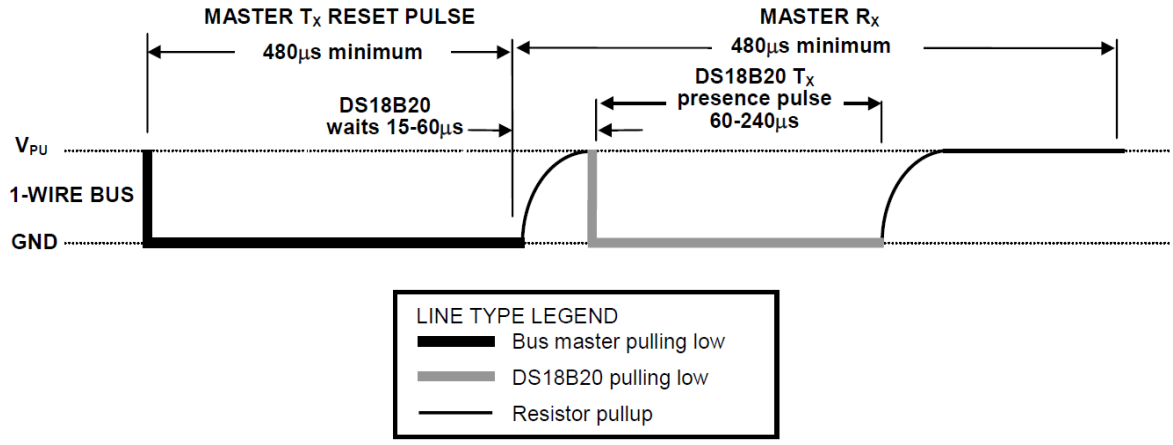
- Standardní – 15,4kbps
- Přetížená (overdrive) – 125kbps

Rozdíl mezi rychlostmi je v tom, že u standardní rychlosti je délka jednoho time slotu 60 µs a u přetížené rychlosti je tento interval snížen na pouhých 8µs. Reset a prezenční pulz je také pozměněn a používá se pro přepínání rychlostních režimů. Přetížená rychlost je podporována pouze některými zařízeními.

2.6 Inicializační procedura – Reset a prezenční pulz:

Veškerá komunikace začíná inicializační sekvencí, která se skládá z resetovacího pulzu vyvolaného masterem a prezenčním pulzem vyslaným slavem. Prezenční pulz říká masterovi, že zařízení na sběrnici je připraveno ke komunikaci.

Resetovací pulz master vyvolá připojením sběrnice k zemi po do dobu minimálně 480 µs. Poté master sběrnici uvolní a začne naslouchat. Sběrnice se opět po uvolnění dostane do stavu 1, díky 5k Ω pullup rezistoru. Když slave zpozoruje rostoucí hranu signálu na sběrnici, počká 15-60 µs, a vyšle prezenční pulz uzemněním sběrnice po dobu 60-240µs.



Obrázek 3: Resetovací a prezenční pulz

2.6.1 Write/Read time sloty:

Master přijímá a zapisuje data do slave zařízení pomocí read a write time slotů. Během každého time slotu je přenesen jeden bit.

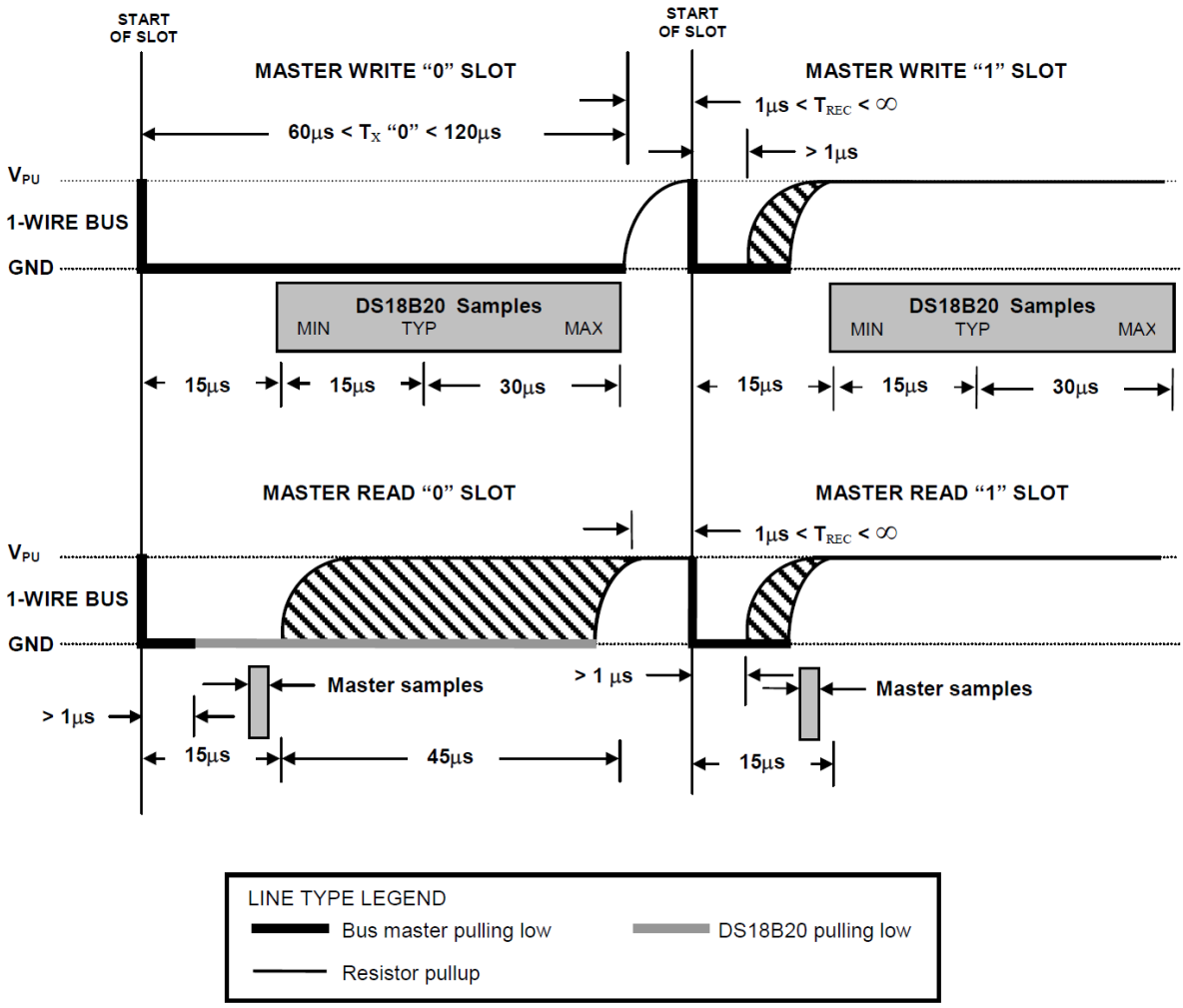
2.6.1.1 Write time sloty:

Máme dva typy write time slotů a to: „Write 1“ time slot a „Write 0“ time slot. Write 1 slouží k odeslání logické jedničky a write 0 k odeslání logické nuly do slave zařízení. Všechny write time sloty musí trvat minimálně 60µs a mezi jednotlivými musí být alespoň 1µs přestávka. Oba typy write time slotů iniciuje master stažením sběrnice k zemi.

Pro write 1 time slot musí master po přizemnění sběrnici do 15µs opět uvolnit. Pro write 0 time slot musí master sběrnici přizemnit po dobu minimálně 60µs. Slave vzorkuje stav sběrnice během doby mezi 15-60µs po započetí write time slotu.

2.6.1.2 Read time sloty:

Slave může vysílat data pouze když master vyvolá read time slot. Každý read time slot musí trvat minimálně 60µs a mezi jednotlivými musí být pauza alespoň 1µs. Read time slot master vyvolá stažením sběrnice k zemi po dobu minimálně 1µs a poté sběrnici uvolnit. Následně slave vyšle 1 nebo 0 na sběrnici. Jednička je vyslána ponecháním sběrnice ve vysoké úrovni a nula přizemněním sběrnice. Data jsou platná po dobu 15µs od započetí read time slotu, a proto je musí master do této doby navzorkovat.



Obrázek 4: Write a Read time sloty

2.6.2 Overdrive komunikace:

Overdrive komunikace využívá stejnou signalizaci jako klasická, jediný rozdíl je v délce jednotlivých intervalů. Více o overdrive komunikaci například v [10] nebo [11].

V kapitole 2 bylo čerpáno z [1] , [8], [9], [10], [11].

3 DS18B20

DS18B20 je digitální teploměr. Ke komunikaci s okolím používá sběrnici 1-wire, a jako každé 1-wire zařízení má svůj jedinečný 64-bitový kód.

3.1 Základní vlastnosti:

- Může být napájen z datového vodiče. Napájení je v rozsahu 3,0V – 5,5V.
- Měří teploty v rozsahu -55°C až +125°C.
- Přesnost ±0,5 °C od -10°C do +85°C.
- Přesnost je uživatelsky volitelná od 9 do 12 bitů.
- Teplota je převedena do 12 bitového slova za maximálně 750ms.
- Uživatelsky definovaný teplotní alarm.
- Příkaz pro vyhledání zařízení, jehož naměřená hodnota teploty překračuje nastavenou hodnotu alarmu.

3.2 Měření teploty:

Základní funkcionalitou DS18B20 je digitální teplotní senzor. Přesnost senzoru je uživatelsky nastavitelná na 9, 10, 11 nebo 12 bitů, které odpovídají násobkům 0,5°C, 0,125°C a 0,0625°C. Po prvním spuštění je teplotní rozlišení nastaveno na 12 bitů. Senzor po přivedení napájení začíná v módu nečinnosti. Pro zahájení teplotního měření, master musí zahájit převod zasláním Convert T příkazu. Následuje měření teploty, která je uložena v 2-bajtovém registru v zápisníkové paměti a následně se senzor opět vrátí do módu nečinnosti. Pokud je DS18B20 napájen externím zdrojem, master může pomocí "read time slots" zjišťovat, zda je již převod teploty hotov (senzor odpovídá nulou, pokud převod není hotov a jedničkou po dokončení převodu). Pokud je senzor napájen pomocí parazitního napájení, nelze této funkcionality využít.

Převedená teplota je ve °C, pokud požadujeme jinou stupnici je třeba provést přepočet. Teplota je rozdělena do dvou bajtů jako na následující ilustraci.

Byte s nižší vahou:

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}

Byte s vyšší vahou:

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
S	S	S	S	S	2^6	2^5	2^4

Kde S signalizuje kladnou (S = 0), nebo zápornou teplotu (S = 1). Pokud je nastavena 11-bitová přesnost, tak je nedefinovaný bit 0. Pro 10-bitovou přesnost je nedefinovaný bit 1 a bit 0 a pro 9-bitovou přesnost bit 2, bit 1 a bit 0. Při 12-bitové přesnosti jsou definovány všechny bity.
Příklady převodu pro 12-bitovou přesnost:
0000 0111 1101 0000 = +125°C
0000 0000 0000 0000 = 0°C
1111 1110 0110 1111 = -25.0625°C

3.3 Využití alarmu:

Po provedení teplotního převodu, je naměřená teplota porovnána s uživatelsky definovanýma hodnotama 1-bajtových registrů T_H a T_L.

T_H a T_L:

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
S	2^6	2^5	2^4	2^3	2^2	2^1	2^0

T_H a T_L jsou stejné, elektricky nezávislé (uložené v EEPROM paměti) registry. Jsou přístupné skrze zápisníkovou paměť, jako druhý a třetí byte v pořadí. Pouze bity 11 až 4 z naměřené teploty jsou porovnány s T_H a T_L, protože jsou to pouze 8-bitové registry. Pokud je naměřená teplota nižší, nebo stejná jako T_L a vyšší než T_H, dojde k nastavení příznaku alarmu. Příznak alarmu je kontrolován při každém měření teploty. Pokud se teplota dostane do přípustných mezí, příznak je zrušen při dalším měření.

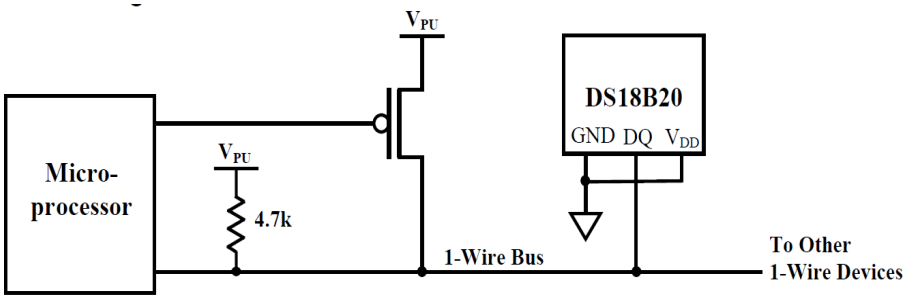
Vedoucí zařízení (master) může ověřit stav každého DS18B20 na sběrnici provedením Alarm Search příkazu. Pokud je alarm nastaven a dojde ke změně hodnot T_H a T_L, je nutné provést nové měření teploty, kdy dojde k novému ověření alarmu.

3.4 Napájení:

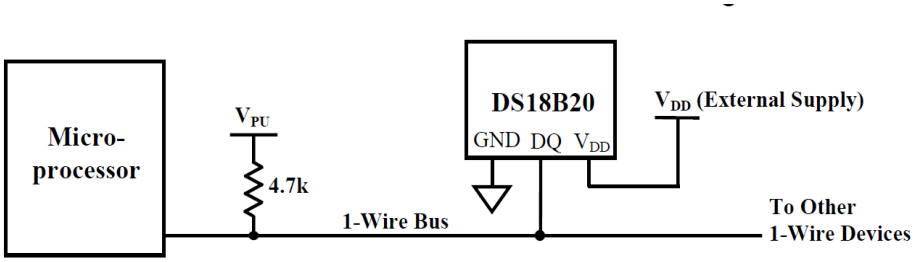
DS18B20 může být napájen externím zdrojem přivedeným na V_{DD} pin, nebo může využívat parazitní napájení, které umožňuje senzoru fungovat bez místního zdroje napájení. Parazitní napájení je velmi užitečné pro aplikace, kde je vyžadováno vzdálené měření teploty, nebo v případě omezeného místa. Při parazitním napájení je napětí odebíráno z datového vodiče, kdy je nastaven v jedničce. Napětí ze sběrnice pohání senzor přímo při jedničce na datovém vodiči a v případě logické nuly se využívá energie uložené v parazitním kondenzátoru uvnitř zařízení. Pokud používáme parazitní napájení, musíme pin V_{DD} připojit na zem. Metoda parazitního napájení zvládne provádět většinu příkazů, krom příkazu k měření teploty a zapsání dat do paměti EEPROM, protože operační proud musí být vyšší

nežli 1,5mA, který není schopen propustit 1-wire pullup rezistor a ani dodat parazitní kondenzátor. Toto se řeší pomocí tvrdého pullup, který je aktivní po dobu provádění konverze teploty (doba jeho zapnutí závisí na zvolené přesnosti), nebo po dobu zápisu do paměti EEPROM (10ms). Pokud je aktivní silný pullup, na sběrnici nelze provádět žádnou další aktivitu.

Pokud použijeme externí napájení, můžeme během vykonávání měření teploty provádět na sběrnici další příkazy.



Obrázek 5: Parazitní napájení



Obrázek 6: Napájení z externího zdroje

3.5 64-bitový ROM kód:

Každý DS18B20 obsahuje unikátní 64-bitový kód uložený v paměti ROM.

8-bit CRC	48-bit sériové číslo	8-bit identifikátor typu (28h)
MSB LSB	MSB LSB	MSB LSB

Nejnižší bajt reprezentuje číslo, které jednoznačně určuje o jaký druh zařízení se jedná. Následuje 48-bitové unikátní sériové číslo. Jako poslední bajt je kontrolní CRC součet předchozích čísel.

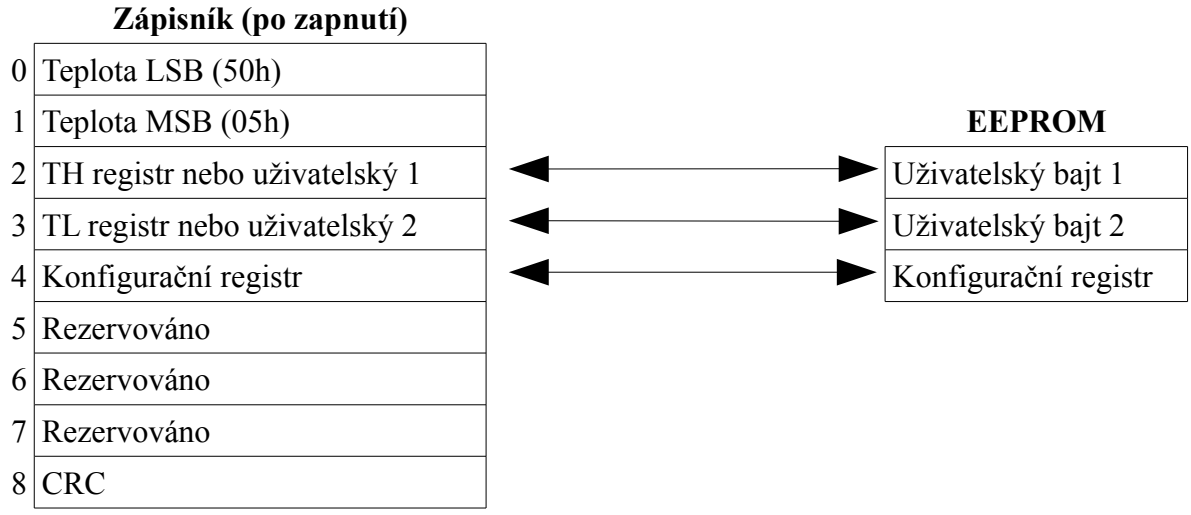
3.6 Paměť:

Paměť se skládá z SRAM zápisníku a napěťově nezávislé EEPROM paměti pro uložení trigovacích registrů (T_H a T_L) a konfiguračního registru. Pokud nepoužíváme funkci alarmu, můžeme registry T_H a T_L využít jako paměť pro svá data.

Bajt 0 a bajt 1 obsahují LSB a MSB teplotního registru. Tyto bajty jsou pouze ke čtení. Bajty 2 a 3 poskytují přístup k T_H a T_L registrům. Bajt 4 obsahuje konfigurační registr. Bajty 5,6 a 7 se využívají pro interní potřeby zařízení a nedá se do nich zapisovat, pouze číst. Poslední 8 bajt zápisníku obsahuje hodnotu CRC pro bajty 0 - 7 v zápisníku.

Data je možné zapsat do bajtů 2, 3 a 4 příkazem Write Scratchpad. Tyto data poté můžeme uložit do EEPROM paměti příkazem Copy Scratchpad. Při zapnutí zařízení se hodnoty z EEPROM nahrají do zápisníkové paměti a pak celý zápisník přečteme pomocí příkazu Read Scratchpad. Pokud změníme data v zápisníku, ale chceme obnovit data z EEPROM a nechceme proto vypínat zařízení, můžeme použít příkaz Recall E2. Master může po zavolání Recall E2 zjišťovat stav nahrávání pomocí read time slotů, kde mu senzor odpovídá nulou, pokud nahrávání ještě probíhá a jedničkou, když jsou již data nahrána.

3.6.1 Schéma paměti:



3.7 Konfigurační registr:

Čtvrtý bajt zápisníku obsahuje konfigurační bajt. Ten je organizován jako je znázorněno na následujícím obrázku.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	R1	R0	1	1	1	1	1

Pomocí bitů R1 a R0 můžeme nastavovat přesnost převodu teploty. Po zapnutí je výchozí nastavení R0=R1=0. Bity 7 a 0 – 4 jsou pouze pro vnitřní účely a není možné je měnit. Možnosti nastavení rozlišení teploměru a doby trvání převodu jsou uvedeny v následující tabulce.

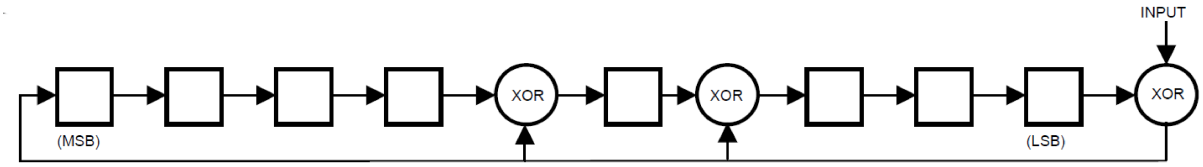
3.7.1 Tabulka rozlišení teploměru:

R1	R0	Rozlišení	Maximální převodní čas	
0	0	9-bit	93,75ms	tconv/8
0	1	10-bit	187,5ms	tconv/4
1	0	11-bit	375ms	tconv/2
1	1	12-bit	750ms	tconv

3.8 Generování CRC:

CRC je součástí 64-bitového ROM kódu a jako 9 bajt zápisníkové paměti. CRC ROM kódu je generováno z prvních 56 bitů a je obsaženo v bajtu s nejvyšší vahou. CRC zápisníkové paměti je počítáno z jejího obsahu a proto je změněno pokaždé, když se data změní. CRC slouží k ověření zda přenos dat proběhl úspěšně. Pro ověření musí master přepočíst CRC z přijatých dat a porovnat hodnotu přijatou s hodnotou vypočtenou.

Výpočet CRC provádí jednoduchý obvod za použití posuvného registru a hradel XOR.



Obrázek 7: Generátor CRC

Na začátku je všech 8 bitů posuvného registru v nule. Bity se postupně přivádějí na vstup (input) od bitu s nejnižší vahou bajtu 0 ze zápisníku a bitu s nejnižší vahou ROM kódu, až po bit s nejvyšší vahou bajtu 7 ze zápisníku a 56 bitu ROM kódu. Nyní se v posuvném registru nachází výsledná hodnota CRC.

3.9 ROM příkazy:

- SEARCH ROM [F0h]

Když je systém zapnut, master musí získat ROM kódy všech slave zařízení na sběrnici, což mu umožní zjistit jejich přesný počet a konkretní typ. ROM kódy získá pomocí procesu eliminace, který se provede vykonáním Search ROM cyklu tolikrát, kolikrát je zapotřebí k zjištění všech zařízení. Po každém Search ROM cyklu se musíme na vrátit k bodu 1 v transakční sekvenci. Pokud je na sběrnici pouze jedno zařízení, můžeme použít jednoduší příkaz Read ROM.

- READ ROM [33h]

Tento příkaz lze použít pokud je pouze jedno slave zařízení na sběrnici. Master tímto příkazem umožněno přečíst celý 64bitový ROM kód bez toho, aniž by musel provést Search ROM proces. Jestliže příkaz použijeme na sběrnici s více slave zařízeními, dojde ke kolizi dat.

- MATCH ROM [55h]

Match ROM příkaz je následován vysláním 64bitového ROM kódu, díky kterému je adresováno konkrétní zařízení na sběrnici. Pouze zařízení jehož ROM kód se plně shoduje s vyslaným ROM kódem bude odpovídat na dále zaslaný funkční příkaz, zbývající zařízení budou čekat na resetovací pulz.

- SKIP ROM [CCh]

Tento příkaz se může použít pro adresaci všech slave zařízení na sběrnici. Například se jako následující funkční příkaz může zaslat Convert T a aktivovat tak převod teploty na všech přítomných DS18B20 zařízeních.

Pří použití s funkčním příkazem, na který slave zařízení odpovídají odesláním dat, nelze použít tento příkaz, když se na sběrnici vyskytuje více zařízení, které jsou schopné na tento příkaz reagovat, jelikož by došlo ke kolizi dat, způsobenou souběžným vysíláním několika zařízení.

- ALARM SEARCH [ECh]

Činnost tohoto příkazu je obdobná jako Search ROM, s rozdílem, že odpovídají pouze slave zařízení s aktivním příznakem alarmu.

3.10 DS18B20 funkční příkazy:

- CONVERT T [44h]

Příkaz inicializuje jedno teplotní měření. Po převodu jsou data uložena v prvních dvou bajtech zápisníkové paměti a DS18B20 se vrátí do nízko odběrového idle režimu. Pokud je DS18B20 napájeno parazitním způsobem, je zapotřebí do 10μs po použití tohoto příkazu aktivovat silný pullup

na 1-wire sběrnici po dobu trvání převodu (tconv). Jestliže je napájení provedeno pomocí externího zdroje, master může pomocí read time slotů zjišťovat průběh převodu. Jestliže DS18B20 odpoví nulou, teplotní převod stále probíhá, jestliže jedničkou, teplotní převod je ukončen. Tato technika nelze použít při parazitním napájení.

- WRITE SCRATCHPAD [4Eh]

Tento příkaz umožňuje zapsat 3 bajty do zápisníkové paměti DS18B20. První bajt je zapsán do registru T_H v zápisníku, druhý do registru T_L v zápisníku a třetí do konfiguračního registru v zápisníku. Data musí být vysílána od bitu s nejmenší vahou a během vysílaní nesmí nastat reset, jinak mohou být data poškozena.

- READ SCRATCHPAD [BEh]

Příkaz slouží k přečtení obsahu zápisníku. Přenos dat začíná od nejméně váženého bitu bajtu 0, až do nejváženějšího bitu bajtu 9. Master může přerušit zasílaní resetem v jakékoliv části přenosu, pokud již má načteny data, která potřebuje.

- COPY SCRATCHPAD [48h]

Příkaz zkopíruje obsah registru T_H, T_L a konfiguračního registru do EEPROM paměti. Pokud je zařízení napájeno parazitně, maximálně do 10μs po odeslání tohoto příkazu musí být aktivován silný pullup na 1-wire sběrnici po dobu minimálně 10ms.

- RECALL E_2 [B8h]

Příkaz nahraje hodnoty T_H, T_L a konfiguračního registru z EEPROM do jim odpovídajících registrů v zápisníkové paměti. Master může pomocí read time slotů vysílaných po tomto příkazu zjišťovat průběh kopírování. Pokud DS18B20 odpoví nulou, kopírování ještě probíhá, pokud jedničkou, tak jsou již data zkopírovaná. Toto zkopírování proběhne automaticky vždy po přivedení napájení k obvodu.

- READ POWER SUPPLY [B4h]

Master po tomto příkazu vyšle read time slot pro zjištění jestli je nějaké zařízení na sběrnici napájeno parazitně. Během read time slotu, parazitně napájení zařízení odpoví nulou a zařízení napájené z externího zdroje ponechá sběrnici v jedničce.

3.11 1-wire signalizace:

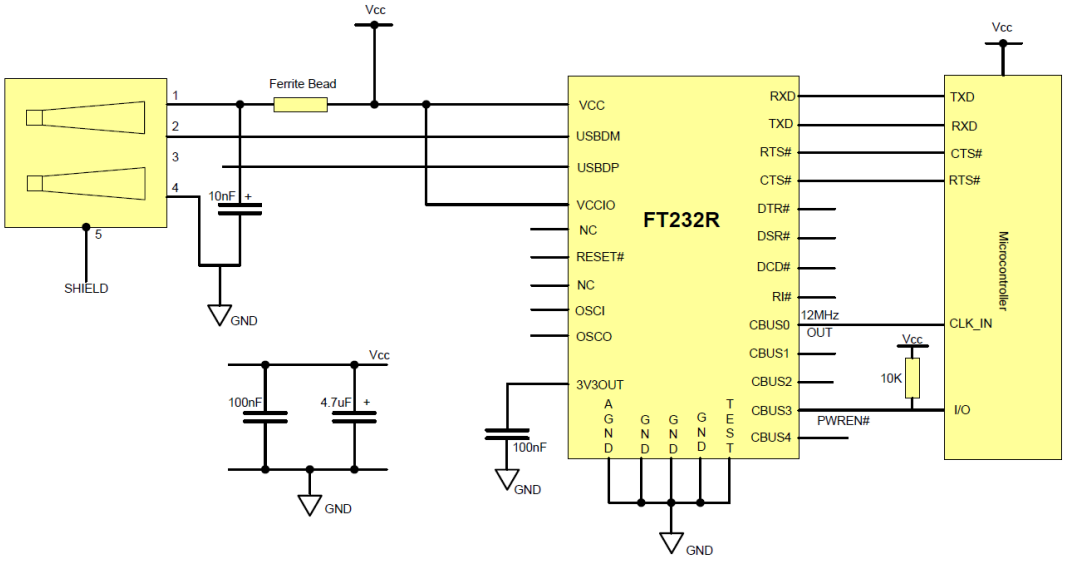
DS18B20 striktně dodržuje 1-wire protokol, aby byla zajištěna datová integrita. V protokolu je definováno několik typů signálu: resetovací pulz, prezenční pulz, zápis nuly, zápis jedničky, čtení nuly, čtení jedničky. Všechny tyto signály, krom prezenčního pulzu, inicializuje master.

V kapitole 3 bylo čerpáno z [1].

4 FT232RL

Převodník mezi RS232 a USB integrovaný v jediném čipu s minimálním množstvím přídavných součástek. Jeho výrobcem je společnost Future Technology Devices International Ltd. (FTDI).

Obvod je zvolen z důvodů relativně snadné použitelnosti a splnění všech požadavků pro danou aplikaci. Obvod se prodává v pouzdře 28 Pin SSOP, které je sice velmi malé, takže pro domácí výrobu plošných spojů více komplikovanější, ale s trochou snahy se dá bezproblémově zapájet. Pro konstrukci 1-wire analyzátoru bylo vycházeno z následujícího výrobcem doporučeného zapojení (Obrázek 8).



Obrázek 8: Doporučené zapojení FT232RL

Bližší specifikace obvodu, jeho další možná použití a doporučená zapojení, elektrické parametry a mnoho dalšího lze nalézt v datových listech tohoto obvodu v dokumentu [2].

4.1 Komunikace s PC:

Výrobce obvodu je nabízena možnost volby mezi dvěma typy ovladačů:

- VCP - Virtual COM port - zařízení se v systému chová jako klasický sériový port, což je vhodné při použití s již napsanou aplikací, kde došlo jenom ke změně rozhraní (RS232 na USB), z důvodů nedostupnosti sériového portu v dnešních moderních počítačích.
- D2XX – Direct driver – ovladač, který umožňuje plně využít možnosti USB sběrnice a jeho API je dostupné pro většinu dnes používaných programovacích jazyků.

Oba ovladače jsou dostupné ve většině operačních systémech, což umožňuje využívat zařízení nezávisle na platformně.
Programem FT_Prog, volně dostupným na stránkách výrobce, můžeme změnit určitá nastavení obvodu, které jsou uložená v paměti EEPROM. Mezi tyto parametry patří například jméno zařízení, nebo sériové číslo a tyto parametry můžeme následně použít k identifikaci v naši aplikaci. Tyto změny můžeme provádět i přímo z naši aplikace, při použití D2XX ovladače.

Ve své aplikaci sem zvolil D2XX ovladač a knihovnu JD2XX pro jazyk Java, která je volně dostupná na internetu [4]. Základní práce s knihovnou je popsána v následujícím textu.
Pro použití jd2xx, musíme nejprve importovat tyto knihovny:

```
import jd2xx.JD2XX;

import jd2xx.JD2XXInputStream;

import jd2xx.JD2XXOutputStream;
```

Následně vytvoříme instanci JD2XX objektu:

```
JD2XX jd = new JD2XX();
```

Nyní získáme seznam všech zařízení připojených k USB a vybere námi požadované, k tomu je několik způsobů získání seznamu:
1.Pomocí seriového čísla

```
Object[] devs = jd.listDevicesBySerialNumber();
```

2.Pomocí jména zařízení

```
devs = jd.listDevicesByDescription();
```

3.Pomocí umístění zařízení

```
devs = jd.listDevicesByLocation();
```

Ze seznamů zařízení vybereme námi požadované, a podle toho jaký seznam jsme použili použijme i vhodnou funkci k připojení zařízení.

```
jd2xx.openByDescription("FT232R USB UART");
```

Před samotným zahájením komunikace musíme ještě nastavit parametry přenosu dat mezi mikroprocesorem a FT232RL.

```
jd.setBaudRate(38400);

jd.setDataCharacteristics(8, JD2XX.STOP_BITS_1, JD2XX.PARITY_NONE);

jd.setFlowControl(JD2XX.FLOW_NONE, 0, 0);

jd.setTimeouts(1000, 1000);
```

Po nastavení už můžeme začít posílat a přijímat data. To můžeme udělat pomocí funkcí read() a write()

```
String msg = "Hello World";
```

```
jd.write(msg.getBytes());
byte[] rd = jd.read(10);
```

nebo pomocí streamu.

```
JD2XXInputStream ins = new JD2XXInputStream(jd);
JD2XXOutputStream outs = new JD2XXOutputStream(jd);
byte[] data = new byte[DATA_SIZE];
int ret = ins.read(data);
outs.write(data);
```

Abychom nemuseli neustále kontrolovat, zda nepřišla nějaká data, můžeme zaregistrovat posluchače, který je zavolán vždy, když nějaká data dostupná.

```
jd.addEventListener(new JD2XXEventListener() {
            @Override
            public void jd2xxEvent(JD2XXEvent event) {
                byte[] data = new byte[DATA_SIZE];
                int ret = ins.read(data);
                for(int i=0; i<ret; i++){
                    data[i];
                    //obsluha přijatých dat
                }
            }
        });
```

a nakonec je třeba vše řádně ukončit

```
ins.close();
outs.jd2xx.close();
outs.close();
```

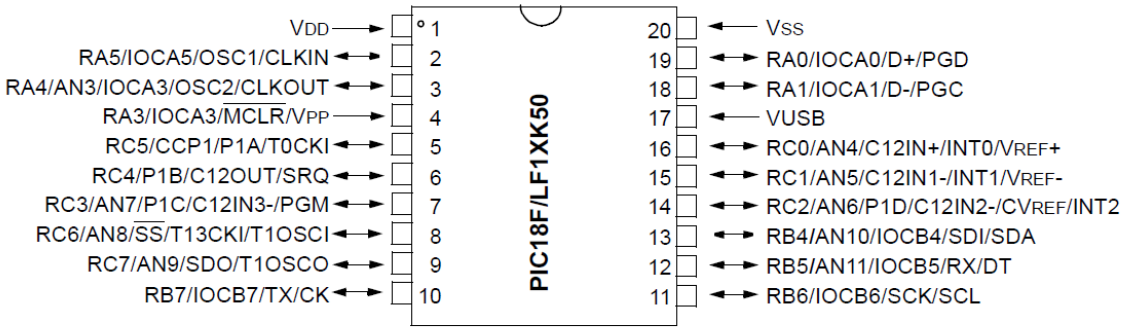
Jak je vidět, práce s JD2XX se podobá komunikaci se sériový portem, ale tato malá ukázka je jen málo z toho, co knihovna JD2XX nabízí, ale pro základní komunikaci je postačující. Více informací lze nálezt v oficiální dokumentaci k D2XX ovladači v dokumentu [3].

V kapitole 4 bylo čerpáno z [2],[3],[4].

5 PIC18F14K50

Základním stavebním prvkem 1-wire analyzátoru je procesor PIC18F14K50 firmy Microchip. Jedná se 8-bitový mikropočítač typu RISC. V procesoru nalezneme 256B paměti EEPROM, 16KB paměti programu a 768B paměti pro data. Dále je vybaven velkým množstvím periferních zařízení, jako jsou 10bitový A/D převodník, EUSART, PWM výstupy, analogové komparátory, I2C, SPI, čtyřmi čítači/časovači, 14 vstupně výstupními porty, řadičem přerušení od mnoha zdrojů a dalšími. Dokáže pracovat s frekvencí až 48MHz, umožňuje používat jak externí, tak nabízí i interní zdroj hodinového signálu. Mnoho podrobných informací, funkcí, možností nastavení lze nalézt opět v datových listech obvodu v dokumentu [5].

Obvod je na trhu dostupný od klasického pouzdra DIP, přes různé SMD varianty až po QFN pouzdro.



Obrázek 9: Rozložení pinů PIC18F14K50 [5]

V aplikaci bude využit hlavně EUSART, a s ním spojená přerušení, jeden 16bitový čítač/časovač v režimu časovač a 4 vstupně/výstupní porty.

5.1 Práce s procesorem:

Pro práci s procesorem, jako je tvorba programu, samotné naprogramování obvodu, ladění programu a další úpravy provádíme v nástroji MPLAB IDE. Jedná se o komplexní vývojové prostředí poskytované firmou Microchip, jehož součástí je i zdařilý simulátor většiny dostupných procesorů, takže náš program můžeme zkoušet, aniž bychom ho nahráli fyzicky do procesoru.

Program se dá napsat jak v assembleru, tak jde využít i vyšší programovací jazyk ansi C, který je přeložen do assembleru například pomocí HI-TECH PICC18 kompilátoru, který je ve verzi Lite volně dostupný. HI-TECH PICC18 nenabízí pouhý překlad, ale obsahuje i knihovny funkcí, které nám

usnadní práci při vývoji. Více o možnostech, syntaxi jazyka, direktiv kompilátoru HI-TECH PICC18 se dočteme v dokumentu [6].

Na začátku každého programu je potřeba nejprve vhodně nastavit periferie, což se provádí změnou konfiguračních registrů procesoru. O tom jak co nastavit se dočteme v již zmiňovaných datových listech tohoto procesoru.

Po napsání aplikace a jejím úspěšném přeložení je zapotřebí ji nahrát do procesoru. Procesory od společnosti Microchip se programují prostřednictvím ICSP (In-Circuit Serial Programing), k čemuž je zapotřebí jeden z mnoha dostupných programátorů, například ICD2. Více o ICSP se dočteme v dokumentu [7]

5.2 Ukázky zdrojového kódu:

Funkce sendData odešle jeden byte na sběrnici 1-wire. Ve funkci se 8x v cyklu zopakuje dotaz: Je-li bit z odesílaného bajtu na i-té pozici (pro i od 0 do 7) roven 1, pak na sběrnici vyšli Write 1 time slot, jinak vyšli na sběrnici Write 0 time slot.

```
void sendData(unsigned char data)

{

      unsigned char counter;
      for(counter=0; counter<8 ;counter++){
                  if(data&(1<<counter)){
                        OUTPUT = 1;
                        __delay_us(6);
                        OUTPUT = 0;
                        __delay_us(100);
                  }
                  else{
                        OUTPUT = 1;
                        __delay_us(100);
                        OUTPUT = 0;
                        __delay_us(10);
                  }
      }
}
```

Funkce readData načte jeden bajt z 1-wire sběrnice. V cyklu se 8x provede následují algoritmus. Na začátku master vyšle 5µs dlouhé uzemnění sběrnice a následně ji uvolní, čímž dostane slave pokyn pro vyslání dat. Master se přepne do režimu příjmu a po 5µs přečte stav sběrnice. Pokud načetl jedničku, nastaví příslušný bit ve výsledku.

```
unsigned char readData()
{
      unsigned char data,counter;
      data=0;
      for(counter=0; counter<8; counter++){
                  OUTPUT = 1;
                  __delay_us(5);
                  OUTPUT = 0;
                  __delay_us(5);
                  if(INPUT){
                        data|=(1<<counter);
                  }
                  __delay_us(70);
      }
      return data;
}
```

Funkce reset slouží k resetu sběrnice, master uzemní sběrnici po dobu 500µs a poté ji uvolní a čeká dalších 500µs až přejde prezenční pulz.

```
void reset()
{
      OUTPUT = 1; //reset
      __delay_us( 500 );
      OUTPUT = 0;
      __delay_us( 500 ); //skip presence pulse checking
}
```

V kapitole 5 bylo čerpáno z [5], [13].

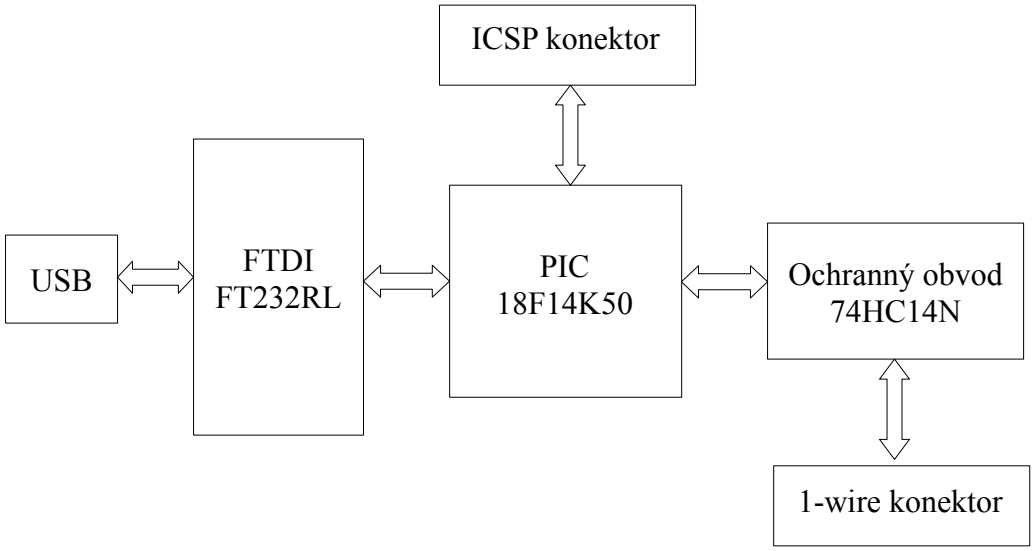
6 Fyzický návrh 1-wire analyzátoru

Jak bylo zmíněno v předešlých kapitolách, stěžejními částmi konstrukce 1-wire analyzátoru je 8-bitový mikroprocesor firmy Microchip PIC18F14K50 a převodník RS232 na USB od firmy FTDI FT232RL.

Návrh schématu zapojení a tištěného spoje byl proveden v programu CadSoft Eagle 6.1.0 Light Edition, která je pro studijní účely k dispozici zdarma.

Schéma zapojení bylo navrženo tak, aby bylo zapotřebí co nejméně přídavných součástek, ale také s ohledem na stabilitu a funkčnost zařízení. Celé zařízení je napájeno 5V, které jsou brány z USB sběrnice, takže zařízení nepožaduje jiný externí zdroj napájení. Na Zařízení je vyveden 5-pinový ICSP konektor, takže není problém aktualizovat firmware, nebo dokonce, sice s omezeními, změnit celý účel zařízení. Druhým konektorem je 3-pinový konektor, který slouží k připojení na 1-wire sběrnici. Třetím a posledním konektorem je USB konektor typu B, který se používá pro připojení k PC pro přenos dat a také pro již zmíněné napájení.

6.1 Blokové schéma:

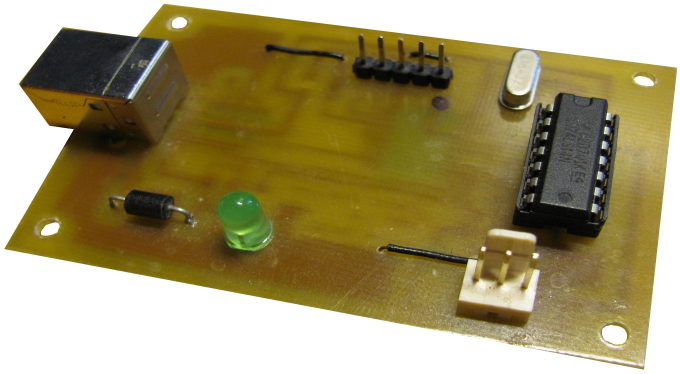


Obrázek 10: Blokové schéma

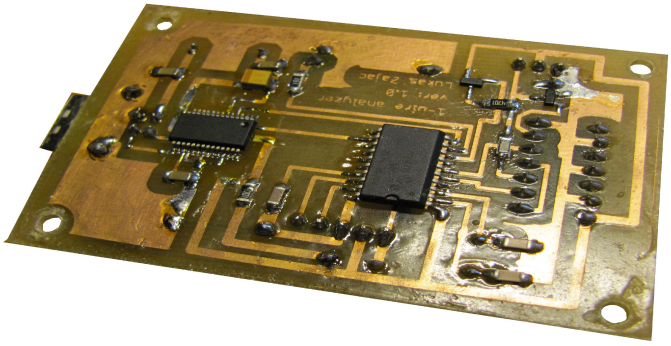
6.2 Výsledné zařízení:

Ze strany součástek (Obrázek 11) se nacházejí všechny 3 konektory, krystal, feritová perla, zelená led dioda a patice pro umístění IC3. Z důvodů snížení složitosti tištěného spoje, jsou zde taktéž umístěny 3 drátové propojky.

Ze strany spojů (Obrázek 12) je umístěna většina součástek. Toto je z důvodů, že velká část součástek je v provedení SMD, které se sice se hůře ručně pájí, ale zato je výsledný produkt menší a stejně se SMD nedá vyhnout, jelikož FT232RL není dostupné v jiném, než SMD pouzdře.



Obrázek 11: Pohled ze strany součástek



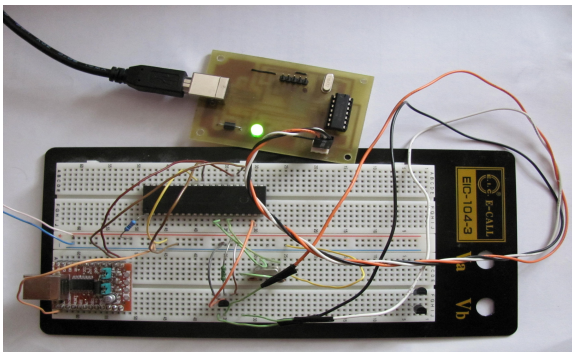
Obrázek 12: Pohled ze strany spojů

Kompletní schémata jsou přiložena na CDROM ve složce *1-wire_board*, kde je i projekt z programu Eagle, aby bylo možné desku snadno upravit nebo rozšířit.

7 Obslužná aplikace

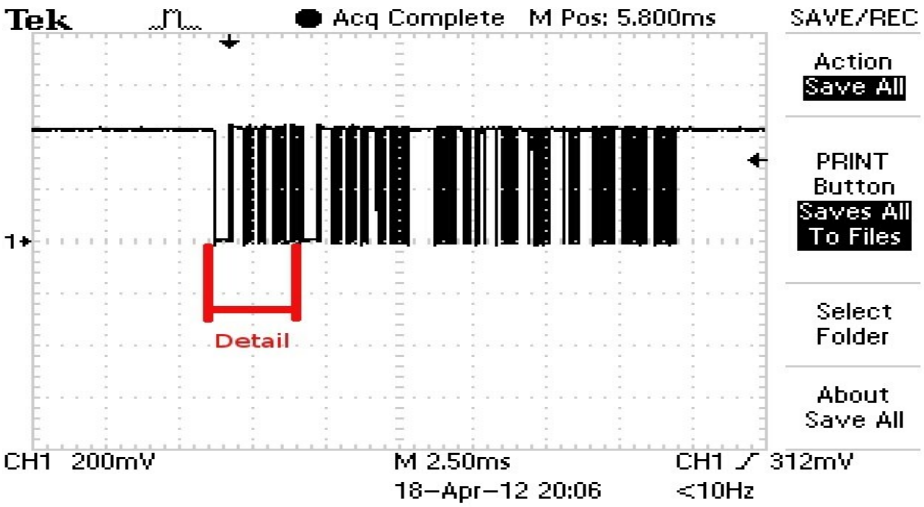
Pro napsání obslužné aplikace byl zvolen programovací jazyk Java. Jako vývojový nástroj je využit volně dostupný nástroj NetBeans IDE ve verzi 7.0.1 a JDK 1.7. Grafické uživatelské rozhraní je vytvořeno s použitím knihovny Swing, která je taktéž součástí prostředí Java.

7.1 Ukázka použití aplikace:



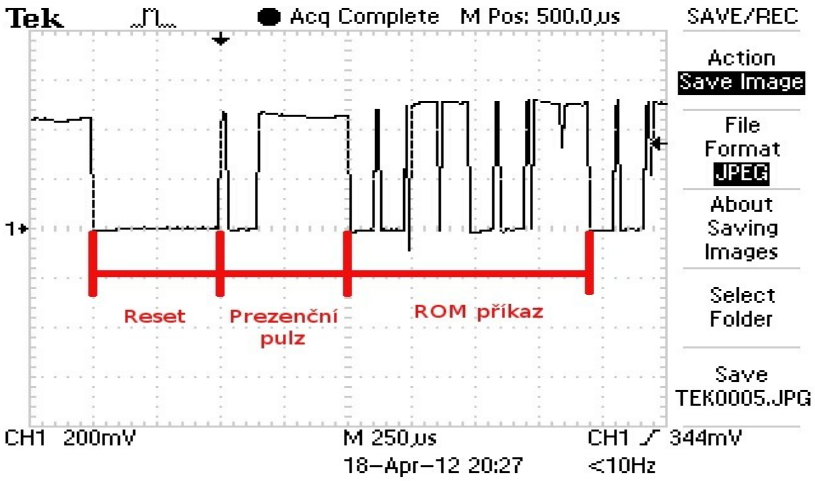
Obrázek 13: Testovací zapojení

Pro účely testování jsem sestavil vlastní zapojení pro simulaci 1-wire komunikace (Obrázek 13), na kterém jsem následně testoval a odlaďoval funkci 1-wire analyzátoru. V procesoru (PIC16F877A) je nahrán program, který periodicky vysílá sestavu příkazů na připojené 1-wire slave zařízení (DS18B20). Na snímku z osciloskopu (Obrázek 14) je celá jedna sestava příkazů .



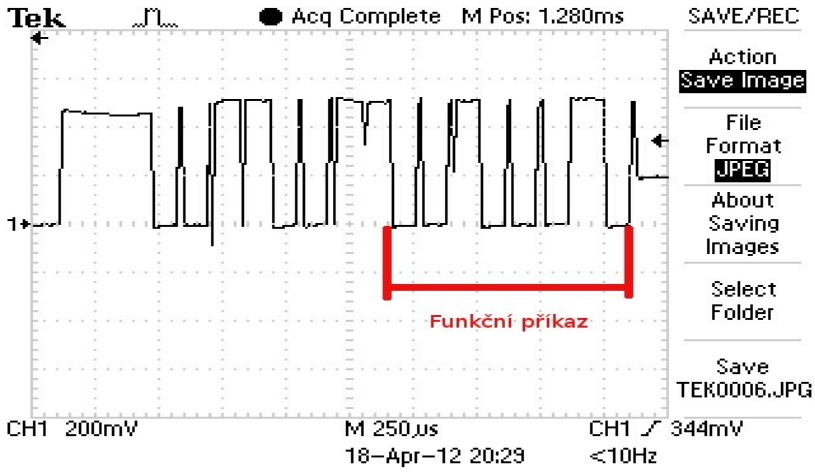
Obrázek 14: Celý průběh signálu

Po přiblížení na začátek jsou již patrné signalizační prvky 1-wire protokolu (Obrázek 15).



Obrázek 15: Detailní pohled na část průběhu

Na začátku průběhu je patrný resetovací pulz s následujícím prezenčním pulzem. Po teto inicializační sekvenci následuje zaslání ROM příkazu. Bity se začínají posílat od bitu s nejnižší vahou, takže zobrazený průběh odpovídá sekvenci 00110011, což po otočení odpovídá bajtu s hodnotou 0b11001100 a to je 0xCC, z čehož vyplývá, že se jedná o Skip ROM příkaz.

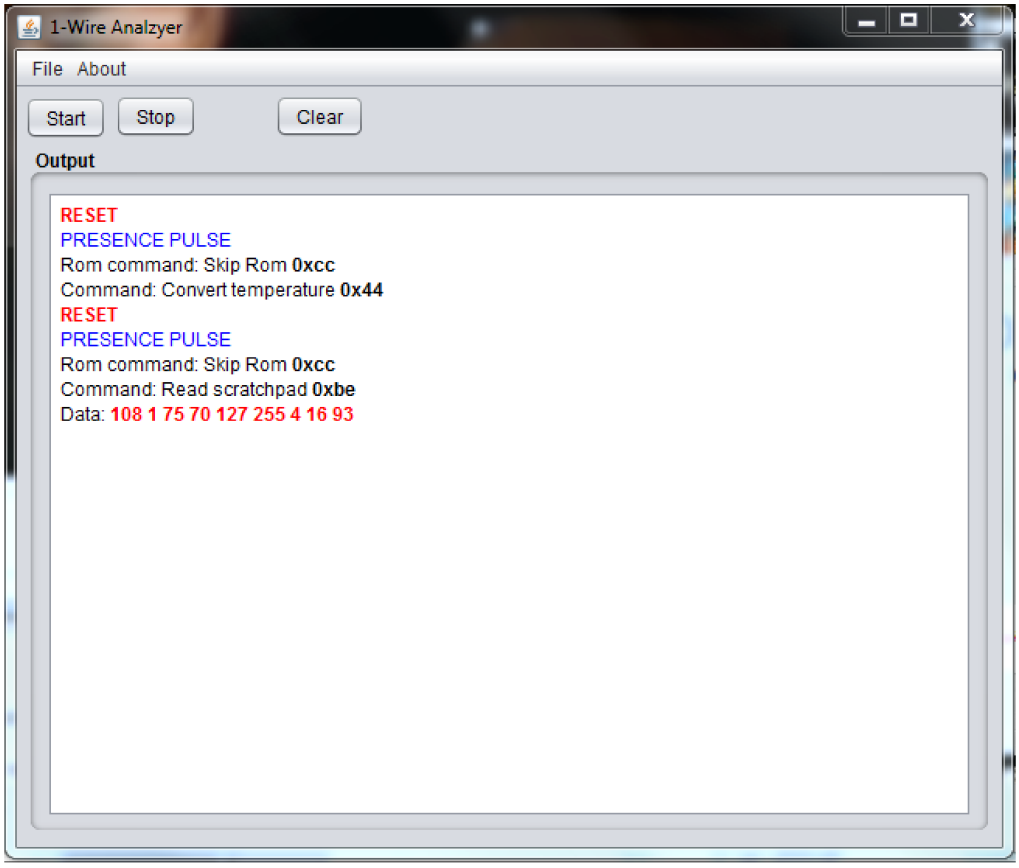


Obrázek 16: Detailní pohled na část průběhu pokračování

Na pokračování průběhu (Obrázek 16) vidíme další celý bajt dat, který by měl podle daného komunikačního protokolu odpovídat funkčnímu příkazu. V bajtu je sekvence 00100010, což je 0b1000100 = 0x44 a to odpovídá funkčnímu příkazu Convert T.

Obslužná aplikace

Nyní je ke sběrnici připojen 1-wire analyzér, na počítači zapnuta obslužná aplikace, v menu zvolen režim 1-wire analyzéru a z nabídky vybráno zařízení, které se na sběrnici nachází (DS18B20). Následným stiskem tlačítka Start byl odchycen průběh komunikace. Jak jde vidět (Obrázek 17), programem byl odchycen Reset, následný prezenční pulz, Skip ROM příkaz a Convert T příkaz.

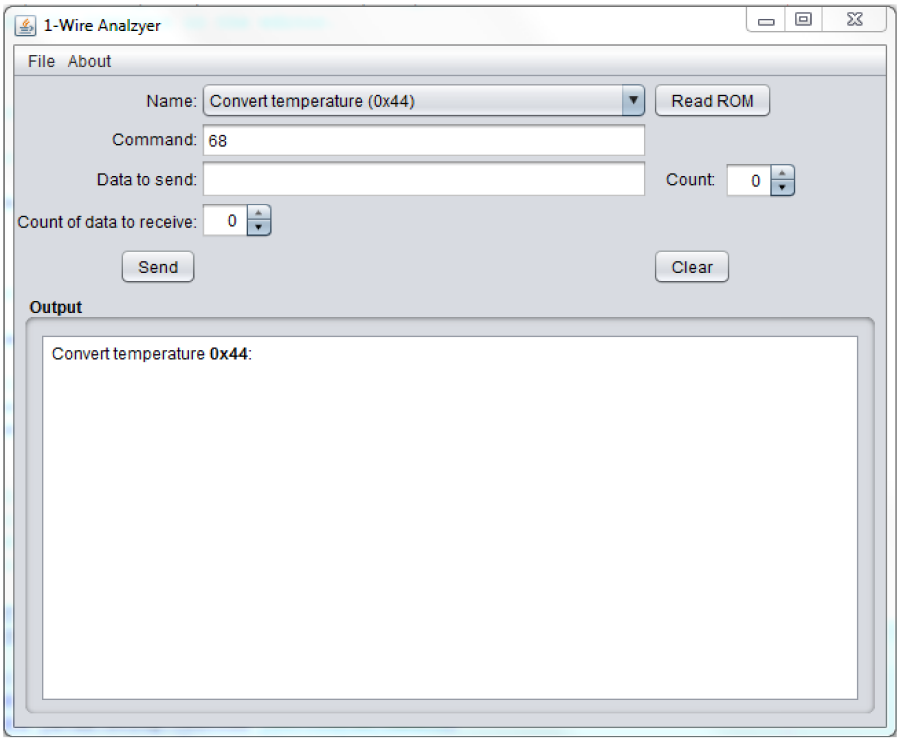


Obrázek 17: Výstup z programu v řežimu 1-wire analyzér

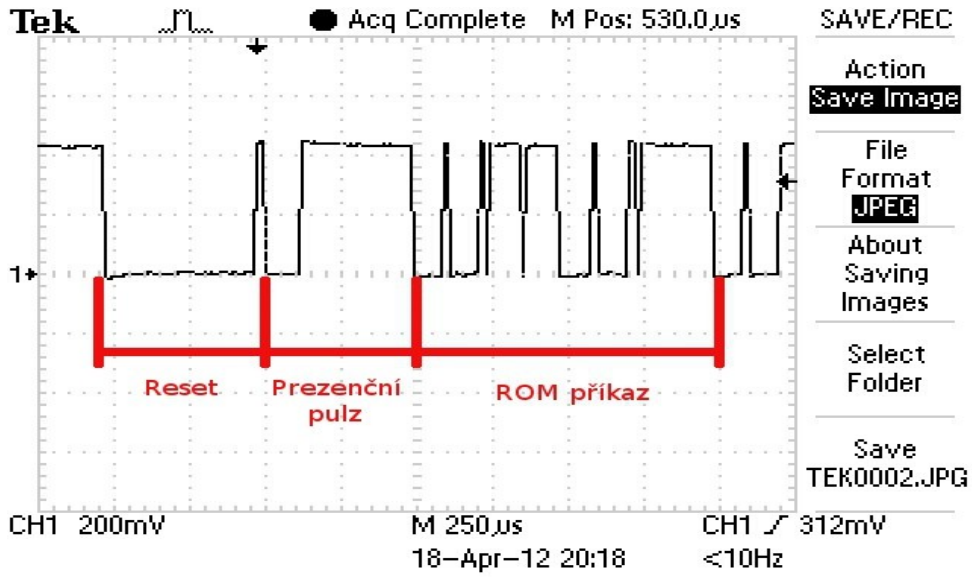
Další částí programu je funkce 1-wire master, jejž funkčnost je ukázána nyní. V aplikaci je zvolen režim 1-wire master, vybráno zařízení, se kterým chceme komunikovat a zvoleno pokračovat. Z nabídky možných příkazů je vybrán Convert temperature a stiskem *Send je* příkaz odeslán (Obrázek 18). Na sběrnici je připojen osciloskop, kterým je odchycen tento datový průběh.

Na Obrázku 19 je vidět začátek průběhu odesílání Convert T příkazu. Počátek průběhu obsahuje, jako každý počátek komunikace na 1-wire sběrnici. reset a prezenční pulz, kterým se master dozví o přítomnosti zařízení a může pokračovat v komunikaci zasláním ROM příkazu. Dále průběh obsahuje 00110011, jelikož na 1-wire sběrnici se bajty odesílají od bitu s nejnižší vahou, je potřeba tuto sekvenci otočit a hodnota odeslaného bajtu je 0b11001100 = 0xCC což odpovídá Skip ROM příkazu.

Obslužná aplikace

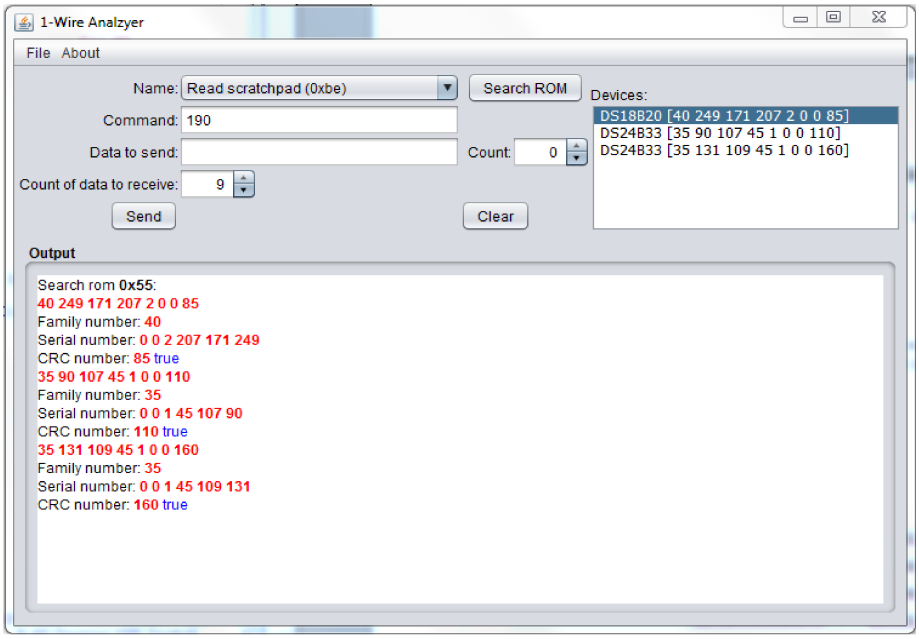


Obrázek 18: Výstup programu v režimu 1-wire master

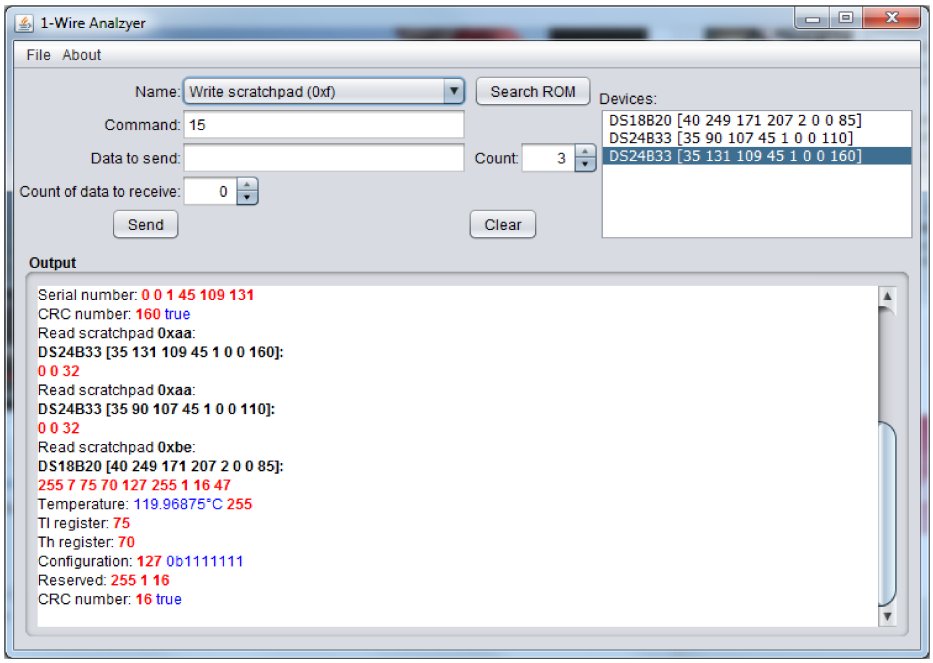


Obrázek 19: Průběh signálu zachycený osciloskopem

Poslední možností programu je komunikace v takzvaném multimódu (multi-drop), kdy je na sběrnici připojeno více zařízení.



Obrázek 20: Stav po vyhledání zařízení na sběrnici



Obrázek 21: Zasílání příkazů na jednotlivá zařízení

7.2 Základní ovládání programu:

Po spuštění aplikace se zobrazí následující okno s nápisem o připojení zařízení.

Obrázek 22: Základní obrazovka

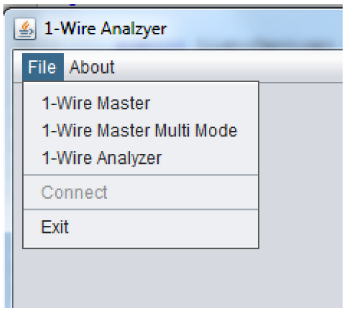
Jestliže před spuštěním aplikace nebylo připojeno zařízení, objeví se chybový nápis jako na Obrázku 23.

Obrázek 23: Orazovka bez připojeného zařízení

Po připojení zařízení do PC, je nutno stisknout tlačítko *Connect* umístěné v menu *File* a jestliže je vše

v pořádku, měl by se změnit nápis na schodný s tím na Obrázku 22 a také se v menu *File* umožní spouštění všech režimů aplikace .

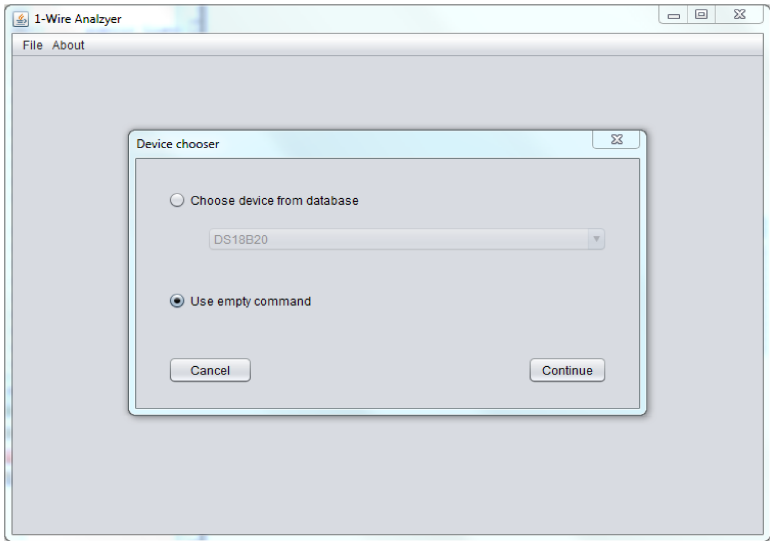
Volba režimu probíhá z již zmíněného menu File (Obrázek 24), kde se dá vybírat mezi *1-Wire Master*, *1-Wire Multimode* a *1-Wire Analyzer*.



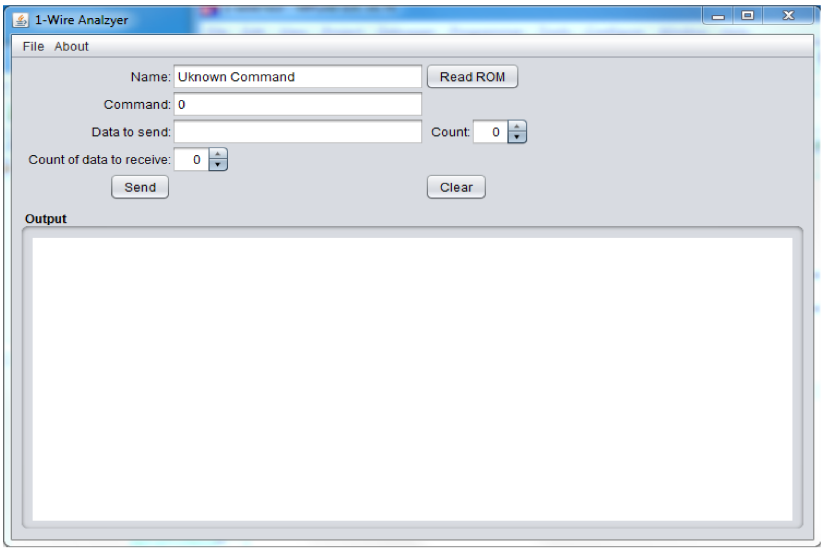
Obrázek 24: File menu

7.2.1 Režim 1-Wire Master:

Při zvolení režimu 1-Wire Master se zobrazí okno (Obrázek 25) s možnostmi, zda se má pracovat jako Empty command, kdy není použit žádný include soubor, ale vše je psáno uživatelem ručně, čili je vhodný pro neznámá zařízení, či pokud je potřeba něco rychle otestovat a uživatel se nechce zaobírat psaním nového includu. Druhá volba je výběr nějakého již známého zařízení z databáze, kdy už jen pak dochází k výběru předvyplněných příkazů a uživatel do nich doplňuje pouze data, která se mají odeslat. Kliknutím na *Cancel* se spuštění zruší a volbou *Continue* dojde již ke spuštění samotného hlavního okna.



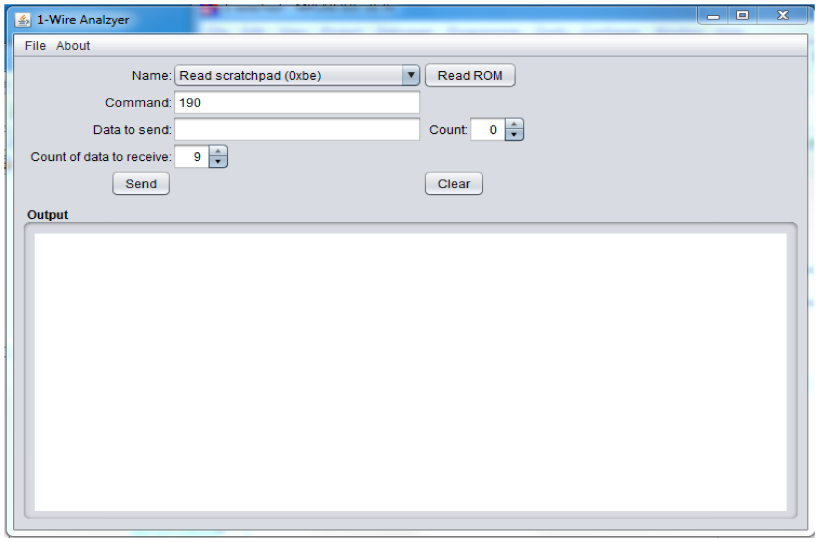
Obrázek 25: Volba zařízení



Obrázek 26: 1-Wire Master - Empty command

Na Obrázku 26 je vidět hlavní okno 1-Wire Master s volbou Empty command. Do položky *Name* se může uvést jméno příkazu. V *Command* se uvádí číselná hodnotu příkazu v hexa soustavě, pokud je zadaný příkaz mimo rozsah 1 - 255, nedojde k jeho odeslání a je u něho znázorněná chyba. *Data to send* obsahuje jednotlivé bajty určené k odeslání v hexa soustavě oddělené čárkami, jejich počet se pro kontrolu uvádí v položce *Count*, kdy dojde před odesláním ke kontrole jejich počtu, aby se zabránilo zbytečným chybám. Poslední editovatelnou položkou je *Count of data to receive*, kde se nastaví kolik bajtů dat se má přijmout. Po vyplnění všech potřebných položek, je příkaz odeslán stiskem tlačítka *Send*, a v okně *Output* se objeví příslušný výstup. Tlačítkem *Clear* dojde k vymazání výstupu a nastavení hodnoty všech vstupních prvků na výchozí. Posledním tlačítko je Read ROM, které složí k přečtení ROM kódu zařízení a vypíše jej na výstup.

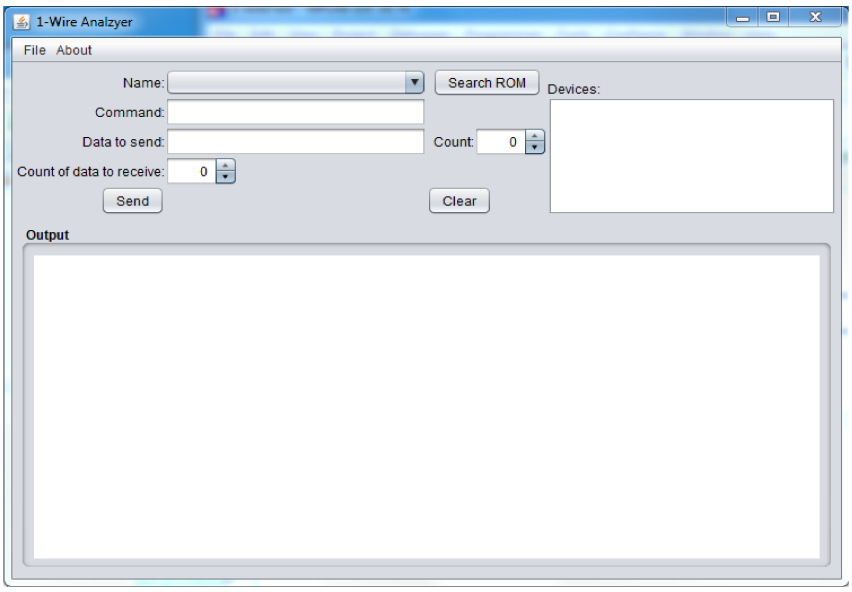
Obrázek 27 ukazuje hlavní okno 1-Wire Master s využitím includu konkrétního zařízení. V položce *Name* se vybírá z nabídky z již napsaných příkazů a jeho hodnoty se přednastaví do zbývajících vstupních prvků. Uživatel pouze doplní *Data to send* stejně jako při volbě Empty command, takže zapisuje jednotlivé bajty v hexa soustavě oddělené čárkami. Jelikož v některých příkazech se může posílat různý počet dat, je jejich počet volitelný změnou natavení *Count*, kdy při odeslání opět dojde ke kontrole, zda počet očekávaných odpovídá počtu vložených bajtů. Stejně tak je možno změnit i položku *Count of data to receive*, čímž je ovlivněn počet přijatých bajtů. Tlačítko *Send* příkaz, pokud neobsahuje chyby, odešle a dojde k zobrazení příslušného výstupu. *Clear* vymaže výstup a obnoví hodnoty vybraného příkazu na výchozí uvedené v include souboru. *Read ROM* přečte ROM kód zařízení a vypíše jej na výstup.



Obrázek 27: 1-Wire Matser – include režim

7.2.2 Režim 1-Wire Multimode:

Režim 1-Wire Multimode umožňuje komunikaci s více než jedním zařízením na sběrnici tkzv. „multi-drop“. Aplikace v tomto režimu umí pracovat pouze se zařízeními, ke kterým existují include soubory. Pokud se na sběrnici vyskytne některé bez include souboru, nebude v nabídce zobrazeno a nepůjde s ním komunikovat. Hlavní okno (Obrázek 28) vypadá obdobně jako hlavní okno 1-Wire Master s využitím include souboru, jenom zde přibyl seznam zařízení na sběrnici.

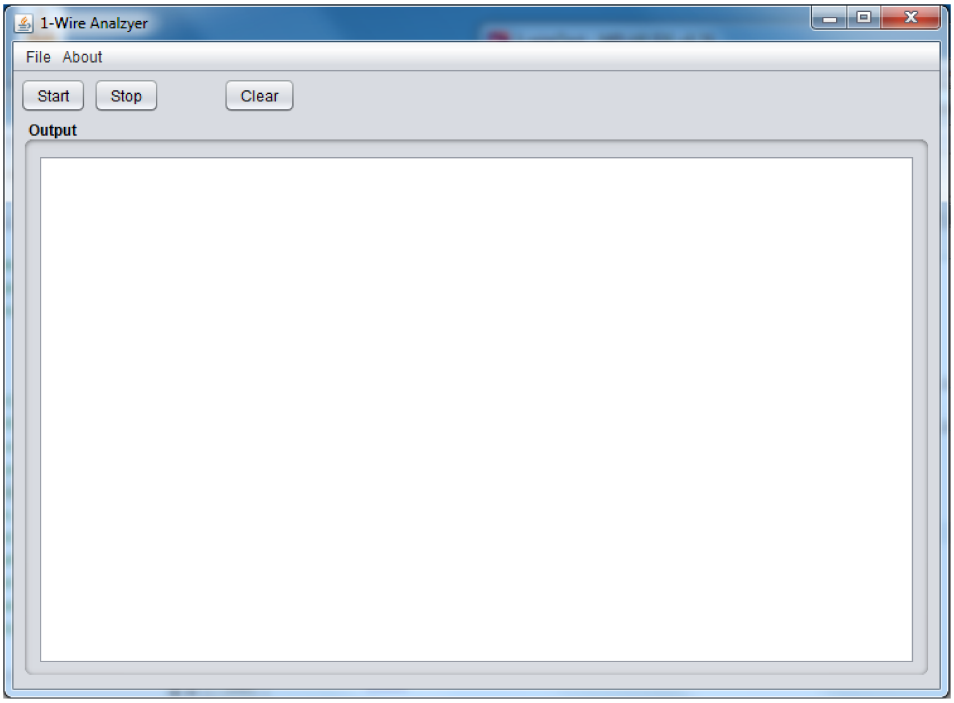


Obrázek 28: 1-Wire Master – Multimode

První krok, který se musí provést, je dát vyhledat všechny zařízení připojené ke sběrnici. To se provede stiskem tlačítka *Search ROM*, což vykoná příslušný výstup a v seznamu zařízení *Device* se objeví všechna nalezená. Poté už jen jen zvoleno zařízení, se kterým se má aktuálně komunikovat a stejným postupem jako s 1-Wire Masterem v include režimu se vybírají příkazy z nabídky, vyplňují data k odeslání (zapisují se v hexa a jednotlivé bajty se oddělují čárkami) a nastavujeme počet dat k přijetí. Příkaz se odešle stiskem *Send*. *Clear* opět vymaže výstup a obnoví změněné parametry aktuálního příkazu na výchozí uvedené v include souboru.

7.2.3 Režim 1-Wire Analyzer:

Posledním dostupným režimem je režim analyzéru, kdy se zařízení připojí k již funkčnímu 1-Wire zařízení a odposlouchává probíhající komunikaci. Po spuštění tohoto režimu se objeví opět nabídka (Obrázek 25), zda se má spustit s includem, nebo bez. Hlavní okno 1-Wire Analyzéru je stejné pro



Obrázek 29: 1-Wire Analyzer

oba režimy. Jediné v čem je rozdíl je následné použití, kdy při použití include souborů budou dekódovány názvy příkazů i v případě použití single-drop, kdy se využívá Skip ROM příkaz a není možné zjistit typ zařízení. Režim bez include se využívá při použití Match ROM příkazů a na základě ROM kódu je v databázi vyhledán potřebný include. Když include neexistuje, příkazy nejsou dekódovány. Tlačítkem *Start* se zahájí odposlech ze sběrnice a zachycená data se zobrazí na výstupu. *Stop* slouží k zastavení zachytávání a stiskem *Clear se* vymaže výstup.

7.3 Věci potřebné k běhu aplikace:

Pro běh aplikace je potřeba mít nainstalované prostředí Java JRE minimálně ve verzi 1.7. U Windows 7 proběhne instalace ovladačů zařízení automaticky po připojení zařízení, u starších verzí je potřeba provést instalaci ovladačů D2XX manuálně. Ovladače se dají získat na stránkách výrobce FTDI, nebo jsou přiloženy na CDROM v sekci *Drivers*.

7.4 Přidání nového zařízení:

Program je možno rozšiřovat o nové zařízení pomocí xml souborů. Tento soubor se nahraje do adresáře *include* v adresáři s programem. Nový soubor je nejlépe tvořit zkopírováním již existujícího a jeho editací.
Xml soubor má následují strukturu:

```
<?xml version="1.0"?>
<device>
	<devicename>DS18B20</devicename>
	<familynumber>120</familynumber>
	<command>
		<commandname>Read scratchpad</commandname>
		<comandvalue>190</comandvalue>
		<datatosend>0</datatosend>
		<datatoreceive>9</datatoreceive>
		<comandtype>temperature</comandtype>
	</command>
	<command>
		<commandname>Convert temperature</commandname>
		<comandvalue>68</comandvalue>
		<datatosend>0</datatosend>
		<datatoreceive>0</datatoreceive>
		<comandtype>none</comandtype>
	</command>
	<command>
		<commandname>Write scratchpad</commandname>
```

```
                <comandvalue>78</comandvalue>
                <datatosend>3</datatosend>
                <datatoreceive>0</datatoreceive>
                <comandtype>none</comandtype>
        </command>
        <command>
                <commandname>Copy scratchpad</commandname>
                <comandvalue>72</comandvalue>
                <datatosend>0</datatosend>
                <datatoreceive>0</datatoreceive>
                <comandtype>none</comandtype>
        </command>
        <command>
                <commandname>Recall eeprom</commandname>
                <comandvalue>184</comandvalue>
                <datatosend>0</datatosend>
                <datatoreceive>0</datatoreceive>
                <comandtype>none</comandtype>
        </command>
</device>
```

`<devicename></devicename>` - zde se zapisuje jméno zařízení

`<familynumber></familynumber>` - zde se uvádí číslo určující druh zařízení, jelikož hodnota se ve většině datasheetu udává v hexadecimální soustavě, tak aby jsme nemuseli tuto hodnotu při opisování přepočítávat, zapisuje se také v hexadecimální soustavě

`<command></command>` - v tagu command se nacházejí informace o jednom funkčním příkazu zařízení, v tagu command jsou obsaženy následující tagy:

`<commandname></commandname>` - jméno příkazu, textový název

`<comandvalue></comandvalue>` - hodnota příkazu v hexadecimální soustavě

`<datatosend></datatosend>` - počet dat která se musejí nebo mohou odeslat v desítkové soustavě

`<datatoreceive></datatoreceive>` - počet dat k přijetí od zařízení v desítkové soustavě

`<comandtype></comandtype>` - druh příkazu, může nabývat hodnot none (nemá žádný typ), nebo temperature, který jde použít u zařízení DS18B20 při příkazu Read Scratchpad a u jiných zařízení se stejnou strukturou zápisníku.

Potřebné informace o zařízení nalezneme v datových listech. Mezi příkazy se nezadávají ROM příkazy, protože ty jsou již zahrnuty v programu, ale zadávají se zde pouze funkční příkazy.

Po vytvoření soubor uložíme s názvem obsaženým v tagu `<devicename></devicename>` s příponou **.xml* do adresáře *include* a na závěr ještě přidáme záznam do souboru *DeviceList.txt* umístěným tamtéž. Textový soubor *DeviceList.txt* má následující strukturu. Na každém řádku je uvedeno jméno zařízení, které odpovídá názvu_souboru.xml a za ním je po oddělení středníkem uvedeno číslo uložené v `<familynumber></familynumber>`, takže obsah souboru vypadá následovně:

```
DS18B20;28
DS24B33;23
```

Tímto krokem je přidání nového zařízení hotovo a s novým spuštěním aplikace již bude možno ho plně využívat.

8 Závěr

Cílem práce bylo vytvořit Analyzátor sběrnice 1-wire, což obnášelo návrh hardwarového přípravku, vytvoření aplikace běžící v mikropočítači a tvorbu aplikace pro PC, odkud je řízená práce hardwarového modulu a jsou ji zpracovávaná data přijatá modulem.

Navržený hardwarový modul splňuje všechny požadavky pro svou funkčnost a je tedy plně použitelný pro analyzování a komunikaci s 1-wire zařízeními. Požadavek na USB, jako komunikační rozhraní mezi PC a hardwarovým modulem je také dodržen. Výsledná obslužná PC aplikace je plně funkční a při jejím návrhu byl kladen důraz na jednoduchost ovládaní a uživatelskou přívětivost.

Do budoucna by se mohla aplikace dále rozšiřovat o podporu nově vzniklých zařízení, případně po delším používání a více zpětných vazeb od uživatelů vylepšit případné nedostatky, nebo doplnit nové funkcionality, které by mohli zkvalitnit a rozšířit možnosti využití tohoto zařízení.

Seznam použité literatury

[1] MAXIM INTEGRATED PRODUCTS. *DS18B20*: *Programmable Resolution 1-Wire Digital Thermometer* [online]. 2008. vyd. [cit. 2012-04-20]. Dostupné z: http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf

[2] FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD. *FT232R USB UART IC* [online]. 2.10. [cit. 2012-04-20]. Dostupné z: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf

[3] FUTURE TECHNOLOGY DEVICES INTERNATIONAL LTD. *D2XX Programmer's Guide* [online]. 1.3. [cit. 2012-04-20]. Dostupné z: http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX_Programmer's_Guide(FT_000071).pdf

[4] Jd2xx - A Java Native Interface port of the FTDI D2XX USB driver. [online]. [cit. 2012-04-20]. Dostupné z: http://bleyer.org/jd2xx/

[5] MICROCHIP TECHNOLOGY INC. *PIC18F/LF1XK50*: *20-Pin USB Flash Microcontrollers with nanoWatt XLP Technology*[online]. 2010 [cit. 2012-04-20]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/41350E.pdf

[6] MICROCHIP TECHNOLOGY INC. *HI-TECH C Compiler for PIC18 MCUs* [online]. 2011 [cit. 2012-04-20]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/PICC_18_9_80_manual.pdf

[7] MICROCHIP TECHNOLOGY INC. *In-Circuit Serial Programming™ (ICSP™) Guide* [online]. 2003 [cit. 2012-04-20]. Dostupné z: http://ww1.microchip.com/downloads/en/devicedoc/30277d.pdf

[8] 1-Wire. In: *Wikipedia*: *the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2012 [cit. 2012-04-20]. Dostupné z: http://en.wikipedia.org/wiki/1-Wire

[9] 1-Wire Devices - Maxim. MAXIM INTEGRATED PRODUCTS. [online]. [cit. 2012-04-20]. Dostupné z: http://www.maxim-ic.com/products/1-wire/

[10] 1-Wire Tutorial Presentation. MAXIM INTEGRATED PRODUCTS. [online]. [cit. 2012-04-23]. Dostupné z: http://www.maxim-ic.com/products/1-wire/flash/overview/index.cfm

[11] MAXIM INTEGRATED PRODUCTS. *DS24B33*: *1-Wire 4Kb EEPROM* [online]. 2012 [cit. 2012-04-23]. Dostupné z: http://datasheets.maxim-ic.com/en/ds/DS24B33.pdf

[12] 1-Wire Search Algorithm. MAXIM INTEGRATED PRODUCTS. [online]. 2002 [cit. 2012-04-23]. Dostupné z: http://www.maxim-ic.com/app-notes/index.mvp/id/187

[13] MATOUŠEK, David. *C pro mikrokontroléry PIC*: *Práce s mikrokontroléry PIC18F452 a PIC18F1220 v jazyce C*. 1. vyd. BEN - technická literatura, 2011. ISBN 978-80-7300-413-2.

Přílohy

I. Příloha na CD

Obsah CD:

- Zdrojové kódy programu ve formě projektu v programu NetBeans IDE.
- Zdrojový kód pro mikroprocesor ve formě projektu programu MPLAB IDE.
- Zdrojový kód pro mikroprocesor ve formě projektu programu MPLAB IDE použitý v testovacím zapojení.
- Návrh tištěného spoje a schéma zapojení ve formě projektu v programu Eagle.
- D2XX ovladače.
- Datasheety použitých obvodů ve formátu pdf.
- Aplikace 1-Wire Analyzer.
- Detailní fotografie zařízení.
- Seznam součástek a osazovací plán ve formátu pdf.